Contents

List of Figures

Acronyms

EVM Ethereum Virtual Machine. 12

Glossary

51% attack A 51% attack refers to an attack on a Proof-of-Work (PoW) blockchain where an attacker or a group of attackers gain control of 51% or more of the computing power or hash rate. PoW is a system of consensus used by blockchains to validate transactions.. 15

cryptocurrency A cryptocurrency, crypto-currency, or crypto is a digital asset designed to work as a medium of exchange wherein individual coin ownership records are stored in a ledger existing in a form of a computerized database using strong cryptography to secure transaction records.. 11

ETH ETH is a cryptocurrency. It is scarce digital money that you can use on the internet, It's the currency of Ethereum apps... 15

Abstract

This research aims to explore one very promiment and potential application for the blockchain, which is a decentralized electronic voting system, by creating a minimum viable decentralized voting application, capable of launching an election, casting votes and displaying results, all while ensuring transaprency, anonymity, security and above all the correctness of the results.

Keywords: Decentralized Electronic Voting System, Voting, Blockchain

Chapter 1

Introduction

1.1 Background

As the hype surrounding bitcoin is slowly waning, the industry is becoming more and more interested in its underlying technology: the blockchain. Now, proponents claim blockchain technology to be one of the most important new technologies of our time, about to take society by storm, and bitcoin served as the first, most thorough and complete proof of concept. In fact, Marc Andreessen, founder of VC firm Andreessen Horowitz and one of the most influential members of Silicon Valley, claimed in a New York Times article that the invention of the blockchain is as important and influential as the creation of the Internet itself[7].

Along with assets registry, secure sharing of medical data and supply chain and logistics monitoring, electronic voting is one of the most prominent potential applications for the surging blockchain technology.

1.2 Key Concepts

1.2.1 Voting

According to Oxford English dictionary, voting is *"a formal indication of a choice between two or more candidates or courses of action, expressed typically through a ballot or a show of hands"*, and the very first forms of voting date back to approximately 508 B.C in ancient Greece where the earliest form of democracy was implemented[5]. Greeks had a "negative" election – that is, each year voters, who were the male landowners, were asked to vote for the political leader or "candidates" they most wanted to be exiled for the next ten years.

The early ballot system was voters wrote their choice on broken pieces of pots, ostraka in Greek, and from this name comes our present word to ostracize. If any ”candidate” received more than 6,000 votes then the one with the largest number was exiled. If no politician received 6,000 votes then they all remained. Since voters were only male landowners, the number of voters was small. If there was a fairly even spread of votes, no one would be exiled, so usually, only very unpopular political leaders were ostracized or exiled.

The election is the backbone of modern democratic societies, it often takes place at a polling station; it is voluntary in some countries, compulsory in others, such as Australia.

1.2.2 Blockchain technology

Blockchain is a distributed database that maintains a secure and ever-growing ledger of records (known as blocks), allowing for the creation of a system of recording information in a way that makes it difficult or impossible to change, hack, or cheat the system. A blockchain is essentially a digital ledger of transactions that is duplicated and distributed across an entire network of computers.

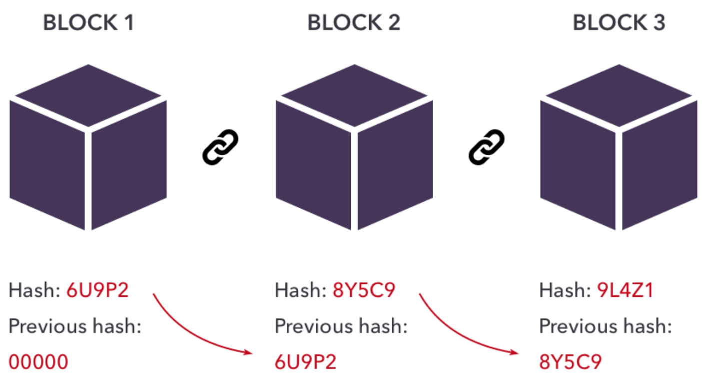

Figure 1.1: Abstract Blockchain blocks visualization

A blockchain network can track orders, payments, accounts, production and much more. And because members of the network share a single view of the truth, you can see all details of a transaction end to end, giving you greater confidence, as well as new efficiencies and opportunities.

1.3 Problem & motivation

Elections are expensive, complicated to set up and even on extreme cases fraudulent, this democratic system is almost 3000 years old and it has not evolved since ancient Greece. It entails an obligation for the voters to rely on third party officials and groups to safely collect votes and publish results, In the case of the modern democratic nation of France for example, there are only 3000 magistrates delegated to supervise and monitor more than 35 000 communes[1], a feat that is inconceivable for them to physically achieve, so setting the human malice apart still leaves room for human error, both leading to the voters carrying justifiable suspicions to any result the ballot may yield. Classic elections also carry hefty financial burdens, the 43rd general election (December 31st, 2020) of Canada had a total estimated cost of $502.4 million[4], the equivalent of the cost of building 5 new hospitals, including administrative areas, operating and emergency rooms, and space for 120 beds[2].

Figure 1.2: The Votomatic vote recorder, a punch card voting machine originally developed in the mid 1960s

Then comes technology, the premise held while entering various fields is that it will increase efficiency, cut costs and leave all parties involved satisfied, with an abundance of examples of when technology did also keep that premise, voting seems to be a logical place to land in next. Electronic voting systems and semi-electronic ones existed for quite some time, from punched card systems (see figure 1.2) to modern-day digital voting machines. However, they only served to optimize the physical act of punching a card or putting a piece of enveloped paper in a box, although that did cut some costs, may be made the experience of voting more pleasant, but it introduced more security concerns and most importantly it did not solve the essential problem of eliminating third parties, because people still had to trust the hardware and software engineers that programmed the machines to cast the votes, and the servers to not allow for any tampering with the data. People must have faith in not only these people's integrity but also competence.

1.4 Purpose & delimitations

The goal of this research is to investigate the possibility of using the new surging blockchain technology as the underlying technology powering an electronic voting system, offering the ability to store the results in a decentralized ledger that should grant the solution attributes like immutability and transparency. We intend to achieve our goal by designing a decentralized e-voting system as a proof-of-concept, capable of launching an election, casting votes, and displaying results, all while ensuring transparency, anonymity, security and above all correctness of the results. It is of major importance to define the scope of this research document and its underlying implementation. First, the implementation of the proof of concept is not by any means complete and thorough but instead functionality was compromised in favor of delivering an actual election on the blockchain.

Second, regarding blockchain technology, this document will not dive deep into the inner working of the technology, topics like hash functions, cryptography and proof of work mechanism will not be presented in a manner that debates their efficiency since we judge the scientific literature on these manners is of abundance, the focus will instead fall on how the application of this technology on voting would look like. Also as Asaf Ashkenazi said *"Every device and system is hackable—it's just a matter of time and hacker motivation."*[3] so our focus will not fall on defending the robustness of our implementation or any future implementation but rather it will fall on highlighting the promise that this technology presents in terms of robustness. Lastly, code snippets that are unique to this project are included.

VĒRUM

Figure 1.3: Proof-of-concept "Verum" Logo

1.5 Document structure

This document is presented in 4 chapters, starting with the introductory chapter in which we will present the reader with a bit of background of the topic and then delve into formally defining the problem we intend to tackle, followed by a brief description of what lies beyond the scope of this research.

In the second chapter, the reader will be presented with sufficient technical background about the two major fields involved in making this project come to life, being the Ethereum blockchain and Web applications. The next chapter will be about the implementation of our proof-of-concept, a chapter in which tools will be introduced and results will be exposed using diagrams, screenshots, and plain old language.

Finally, in the fourth and last chapter, the results and insights gained through the journey of making our proof-of-concept will be discussed, few conclusions drawn and perspectives on what could be enhanced moving forward with this project.

Chapter 2

Technical background

This chapter presents an overview of the technical concepts involved in the realization of this project, namely the Ethereum Blockchain and Web applications

2.1 Ethereum Blockchain

Ethereum is a blockchain platform with its own cryptocurrency, called Ether (ETH) or Ethereum, and its own programming language, called Solidity. As a blockchain network, Ethereum is a decentralized public ledger for verifying and recording transactions. The network's users can create, publish, monetize, and use applications on the platform, and use its Ether cryptocurrency as payment. Insiders call the decentralized applications on the network "dapps."

The intent of Ethereum is to create an alternative protocol for building decentralized applications, providing a different set of tradeoffs that its creators believe will be very useful for a large class of decentralized applications, with particular emphasis on situations where rapid development time, security for small and rarely used applications, and the ability of different applications to very efficiently interact, are important. Ethereum does this by building what is essentially the ultimate abstract foundational layer: a blockchain with a built-in Turing-complete programming language, allowing anyone to write smart contracts and decentralized applications where they can create their own arbitrary rules for ownership, transaction formats and state transition functions.

Figure 2.1: Ethereum Blockchain logo

In the Ethereum universe, there is a single, canonical computer (called the Ethereum Virtual Machine, or EVM) whose state everyone on the Ethereum network agrees on. Everyone who participates in the Ethereum network (every Ethereum node) keeps a copy of the state of this computer. Additionally, any participant can broadcast a request for this computer to perform arbitrary computation. Whenever such a request is broadcast, other participants on the network verify, validate, and carry out (”execute”) the computation. This causes a state change in the EVM, which is committed and propagated throughout the entire network.

2.1.1 Accounts

In Ethereum, the state is made up of objects called ”accounts”, with each account having a 20-byte address and state transitions being direct transfers of value and information between accounts. An Ethereum account contains four fields:

- The **nonce**, a counter used to make sure each transaction can only be processed once
- The account's current **ether** balance
- The account's **contract code**, if present
- The account's **storage** (empty by default)

”Ether” is the main internal crypto-fuel of Ethereum, and is used to pay transaction fees. In general, there are two types of accounts: **externally owned accounts**, controlled by private keys, and **contract accounts**, controlled by their contract code. An externally owned account has no code, and one can send messages from an externally owned account by creating and signing a transaction; in a contract account, every time the contract account receives a message its code activates, allowing it to read and write to internal storage and send other messages or create contracts in turn.

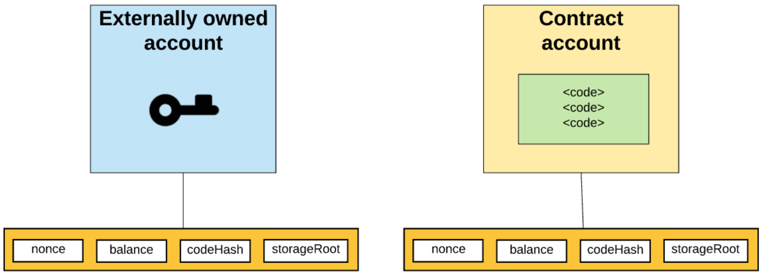

Figure 2.2: Abstract Ethereum accounts visualization

2.1.2 Transactions

An Ethereum transaction refers to an action initiated by an externally-owned account, in other words an account managed by a human, not a contract. For example, if Bob sends Alice 1 ETH, Bob's account must be debited and Alice's must be credited. This state-changing action takes place within a transaction.

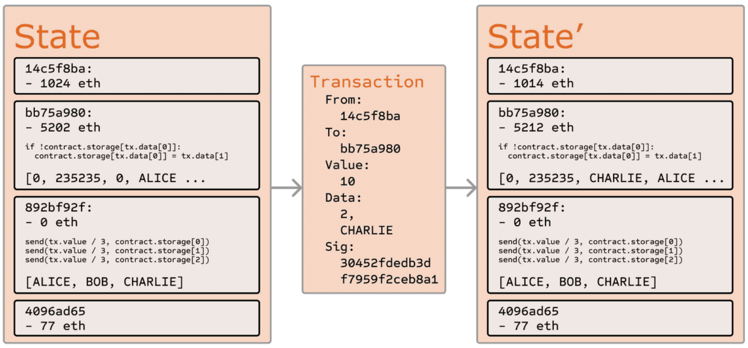

Figure 2.3: Abstract representation of an Ethereum transaction

Transactions, which change the state of the EVM, need to be broadcast to the whole network. Any node can broadcast a request for a transaction to be executed on the EVM; after this happens, a miner will execute the transaction and propagate the resulting state change to the rest of the network.

Transactions require a fee and must be mined to become valid. To make this overview simpler we'll cover gas fees and mining elsewhere.

A submitted transaction includes the following information:

- **recipient** - the receiving address (if an externally-owned account, the transaction will transfer value. If a contract account, the transaction will execute the contract code)

- **signature** - the identifier of the sender. This is generated when the sender's private key signs the transaction and confirms the sender has authorised this transaction
- **value** - amount of ETH to transfer from sender to recipient (in WEI, a denomination of ETH)
- **data** - optional field to include arbitrary data
- **gasLimit** - the maximum amount of gas units that can be consumed by the transaction. Units of gas represent computational steps
- **gasPrice** - the fee the sender pays per unit of gas

Gas is a reference to the computation required to process the transaction by a miner. Users have to pay a fee for this computation. The gasLimit and gasPrice determine the maximum transaction fee paid to the miner (more on gas on later subsections.

2.1.3 Gas

Gas refers to the unit that measures the amount of computational effort required to execute specific operations on the Ethereum network.

Since each Ethereum transaction requires computational resources to execute, each transaction requires a fee. Gas refers to the fee required to successfully conduct a transaction on Ethereum.

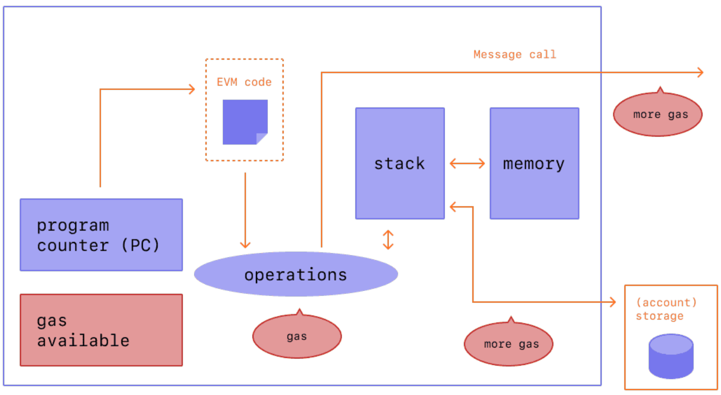

Figure 2.4: Diagram of the role of gas in Ethereum [6]

In essence, gas fees are paid in Ethereum's native currency, ether (ETH). Gas prices are denoted in gwei, which itself is a denomination of ETH - each gwei is equal to 0.000000001 ETH (10-9 ETH). For example, instead of saying that your gas costs 0.000000001 ether, you can say your gas costs 1 gwei.

Gas fees exist because they help keep the Ethereum network secure. By requiring a fee for every computation executed on the network, we prevent actors from spamming the network. In order to prevent accidental or hostile infinite loops or other computational wastage in code, each transaction is required to set a limit to how many computational steps of code execution it can use. The fundamental unit of computation is ”gas”.

2.1.4 Consensus mechanisms

When it comes to blockchains like Ethereum, which are in essence distributed databases, the nodes of the network must be able to reach agreement on the current state of the system. This is achieved using consensus mechanisms.

Consensus mechanisms (also known as consensus protocols or consensus algorithms) allow distributed systems (networks of computers) to work together and stay secure.

For decades, these mechanisms have been used to establish consensus among database nodes, application servers, and other enterprise infrastructure. In recent years, new consensus protocols have been invented to allow cryptoeconomic systems, such as Ethereum, to agree on the state of the network.

A consensus mechanism in a cryptoeconomic system also helps prevent certain kinds of economic attacks. In theory, an attacker can compromise consensus by controlling 51% of the network. Consensus mechanisms are designed to make this ”51% attack” unfeasible. Different mechanisms are engineered to solve this security problem differently.

Types of consensus mechanisms

Proof of work Proof-of-work is done by miners, who compete to create new blocks full of processed transactions. The winner shares the new block with the rest of the network and earns some freshly minted ETH. The race is won by whoever’s computer can solve a math puzzle fastest – this produces the cryptographic link between the current block and the block that went before. Solving this puzzle is the work in ”proof of work”.

Proof of stake Proof-of-stake is done by validators who have staked ETH to participate in the system. A validator is chosen at random to create new blocks, share them with the network and earn rewards. Instead of needing to do intense computational work, you simply need to have staked your ETH in the network. This is what incentivises healthy network behaviour.

2.1.5 Dapps

A dapp has its backend code running on a decentralized peer-to-peer network. Contrast this with an app where the backend code is running on centralized servers.

A dapp can have frontend code and user interfaces written in any language (just like an app) that can make calls to its backend. Furthermore, its frontend can be hosted on decentralized storage.

Decentralized means they are independent, and no one can control them as a group.

Deterministic they perform the same function irrespective of the environment they are executed.

Turing complete which means given the required resources, the dapp can perform any action.

Isolated which means they are executed in a virtual environment known as Ethereum Virtual Machine so that if the smart contract happens to have a bug, it won't hamper the normal functioning of the blockchain network.

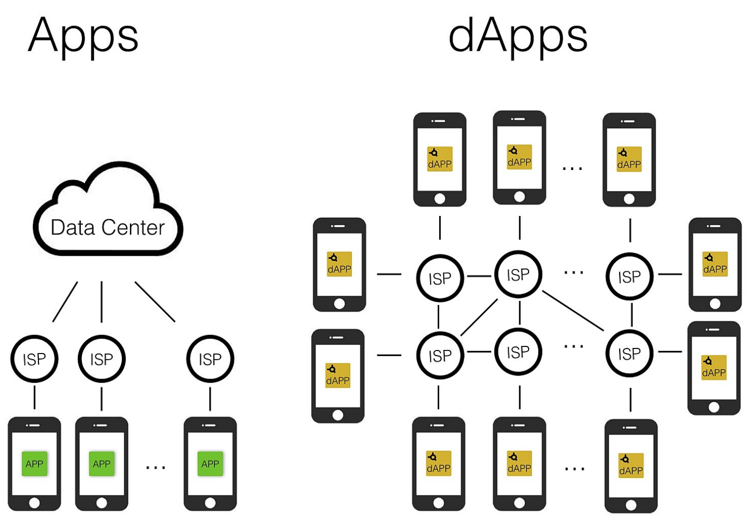

Figure 2.5: Diagram of centralized and decentralized applications topology

2.1.6 Smart contracts

A smart contract is simply put a program that runs on the Ethereum blockchain. It's a collection of code (its functions) and data (its state) that resides at a specific address on the Ethereum blockchain.They runs exactly as programmed. Once they are deployed on the network they can't be changed. Dapps can be decentralized because they are controlled by the logic written into the contract, not an individual or company.

Smart contracts are a type of Ethereum account. This means they have a balance and they can send transactions over the network. User accounts can then interact with a smart contract by submitting transactions that execute a function defined on the smart contract. Smart contracts can define rules, like a regular contract, and automatically enforce them via the code.

Ethereum has developer-friendly languages for writing smart contracts:

- Solidity
- Vyper

2.2 Web Applications

2.2.1 Structure

2.2.2 Business use

2.2.3 Development

Bibliography

[1] Article LO274 - Code électoral - Légifrance. https://www.legifrance.gouv.fr/codes/article_lc/LEGIARTI000006353645.

[2] Cost to Build a Hospital — Hospital Construction Cost. https://www.fixr.com/costs/build-hospital.

[3] Cybersecurity Expert Warns of Increasing Vulnerabilities in Devices — Observer. https://observer.com/2019/09/cybersecurity-expert-asaf-ashkenazi-device-vulnerability-hacking/.

[4] Estimated Cost of the 43rd General Election – Elections Canada. https://www.elections.ca/content.aspx?section=res&dir=rep/off/cou&document=index43&la

[5] History Of Elections. https://www.duvalelections.com/General-Information/Learn-About-Elections/History-Of-Elections.

[6] Takenobu-hs/ethereum-evm-illustrated: Ethereum EVM illustrated. https://github.com/takenobu-hs/ethereum-evm-illustrated.

[7] Marc Andreessen. Why Bitcoin Matters. https://dealbook.nytimes.com/2014/01/21/why-bitcoin-matters/, 1390323270.